

# AVL 树和跳表插入性能测试

杨景凯

520021910550

2022 年 3 月 21 日

## 目录

<b>1</b>	<b>AVL 树随机插入测试</b>	<b>3</b>
1.1	实验说明 . . . . .	3
1.2	实验表格 . . . . .	3
1.3	实验图表 . . . . .	3
<b>2</b>	<b>AVL 树条件插入测试</b>	<b>3</b>
2.1	实验说明 . . . . .	3
2.2	实验表格 . . . . .	4
2.3	实验图表 . . . . .	4
<b>3</b>	<b>结果分析</b>	<b>4</b>
3.1	理论分析 . . . . .	4
3.1.1	实验一 . . . . .	4
3.1.2	实验二 . . . . .	5
3.2	选择 . . . . .	5

## 1 AVL 树随机插入测试

### 1.1 实验说明

此部分针对于 AVL 树进行了不同个数的元素的插入测试。

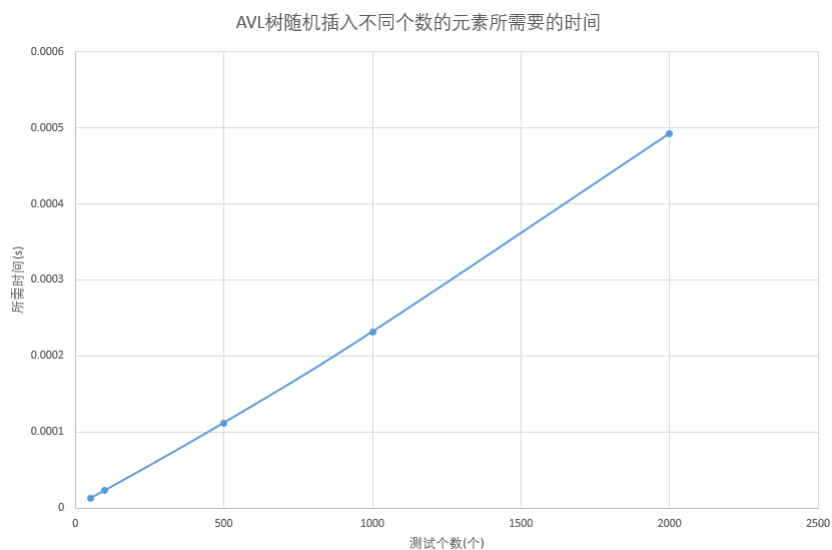
随机生成一组插入序列，插入元素的个数分别为 50, 100, 500, 1000, 2000。

得到了插入元素个数和插入时间的关系。

### 1.2 实验表格

个数	50	100	500	1000	2000
时间	0.000013	0.000023	0.000112	0.000232	0.000493

### 1.3 实验图表



## 2 AVL 树条件插入测试

### 2.1 实验说明

此部分针对 AVL 树与跳表，进行了不同条件的插入测试。

针对插入元素的个数为 2000 的 AVL 树和跳表，对比下列三种情况它们的插入所需时间的差异。

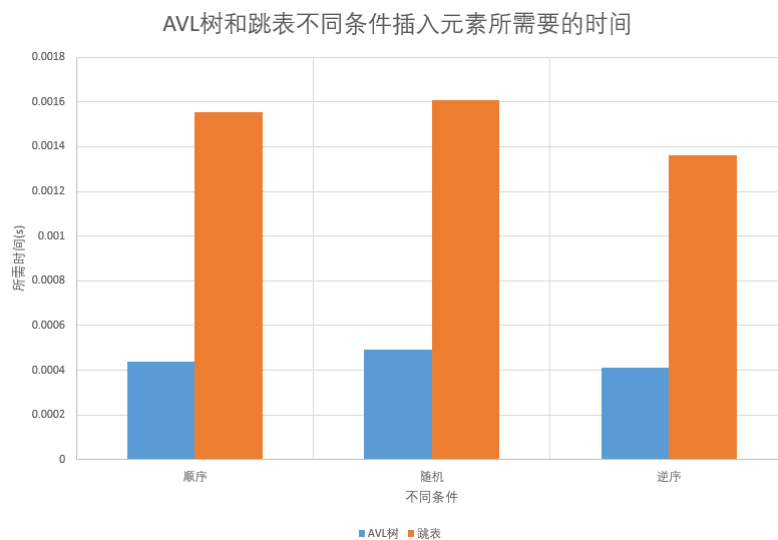
- 1) 顺序插入（后一个插入的 key 恒大于前一个值）
- 2) 随机插入（随机生成插入序列）
- 3) 逆序插入（后一个插入的 key 恒小于前一个值）

得到了两者在不同条件下的插入时间。

## 2.2 实验表格

状况	顺序	随机	逆序
AVL 树	0.000437	0.000493	0.000411
跳表	0.001554	0.001608	0.001361

## 2.3 实验图表



## 3 结果分析

### 3.1 理论分析

#### 3.1.1 实验一

对于 AVL 树来说，插入时间  $T \sim n \log n$ ，因此，随着插入元素增加，插入时间接近于线性增加，略微有上扬的趋势。

通过实验一的表格和图表可以证明这一点，因此实验结果与理论吻合。

#### 3.1.2 实验二

AVL 树对数据排序类型不敏感，因此在顺序，逆序和随机上所用时间差距不大（随机数据有生成随机数的影响）。

跳表对数据排序类型较敏感，逆序比顺序所需时间短，但是差异不大。

通过实验二的表格和图表可以证明这一点，因此实验结果与理论吻合。

但是，理论上跳表和 AVL 树插入时间应该接近，但是实验出现了较为明显的差距。

通过分析，我认为是跳表有计算生成随机数和每个跳表数据块大量数据拷贝的缘故。这导致了跳表要慢于 AVL 树。

### 3.2 选择

在顺序插入大量数据时，我选择 AVL 树。因为此时跳表每次都在末尾插入数据，时间复杂度下界是  $O(n \log n)$ ，而 AVL 树顺序插入时间复杂度为  $O(n \log n)$ 。

在随机插入大量数据时，我认为两个都可以选择，此时两个的平均时间复杂度均为  $O(n \log n)$ 。

在逆序插入大量数据时，我选择跳表。因为跳表每次都可以在开头插入数据，使得平均时间复杂度由  $O(n \log n)$  降为  $O(n)$ ，而 AVL 树顺序插入时间复杂度为  $O(n \log n)$ 。